Volume 5, Issue 3, March 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Hypothetical Authentication Scheme for Enhanced Grid Security

Nirmalya Mukhopadhyay
Asst Prof, Dept of CSE,
B. A. College of Engg & Tech
Jamshedpur, India

Aparna Sen
M. Tech, Dept of CSE
Bankura Unnayani Institute of Engg
West Bengal, India

Subhajit Roy
Asst Prof, Dept of CSE,
Bankura Unnayani Institute of Engg
West Bengal, India

Abstract: Grid computing is concerned with the sharing and use of resources in dynamic distributed virtual organizations. The dynamic nature of Grid environments introduces challenging security concerns that demand new technical approaches. In this brief overview, we have presented a more advanced approach for a new security paradigm, which will ensure the vast use of this technology.

Keywords: Grid Computing, Globus Toolkit, Grid Security, Authentication, Smart Grid, TFMCC, Authenticator.

I. INTRODUCTION

Grid computing[1,2] is a technology for coordinating large scale resource sharing and problem solving among various autonomous group. Grid technologies are currently distinct from other major technical trends such as internet, enterprise distributed networks and peer to peer computing. Also it has some embracing issues in QoS, data management, scheduling, resource allocation, accounting and performance.

Grids are built by various user communities to offer a good infrastructure which helps the members to solve their specific problems which are called a grand challenge problem.

A grid consists of different types of resources owned by different and typically independent organizations which results in heterogeneity of resources and policies. Because of this, grid based services and applications experience a different resource behavior than expected.

Similarly, a distributed infrastructure with ambitious service put more impact on the capabilities of the interconnecting networks than other environments.

Grid High Performance Network Group[3] works on network research, grid infrastructure and development. In their document the authors listed six main functional requirements, which are considered as mandatory requirements for grid applications.

They are:

i) High performance transport protocol for bulk data transfer,
ii) Performance controllability,
iii) Dynamic network resource allocation and reservation,
iv) Security,
v) High availability and
vi) Multicast to efficiently distribute data to group of resources.

Grid computing can mean different things to different individuals. The grand vision is often presented as an analogy to power grids where users (or electrical appliances) get access to electricity through wall sockets with no care or consideration for where or how the electricity is actually generated.

In this view of grid computing, computing becomes pervasive and individual users (or client applications) gain access to computing resources (processors, storage, data, applications, and so on) as needed with little or no knowledge of where those resources are located or what the underlying technologies, hardware, operating system, and so on.

Grid computing could be defined as any of a variety of levels of virtualization along a continuum. Exactly where along that continuum one might say that a particular solution is an implementation of grid computing versus a relatively simple implementation using virtual resources is a matter of opinion. But even at the simplest levels of virtualization, one could say that grid-enabling technologies.

Security is a latest topic today for the smart grid, and progresses are being done in this field every day. Most communications uses standard cryptographic algorithms AES-128 to protect the data on the network. Grid computing is a technique which provides high-performance computing; in this resources are shared in order to improve the performance of the system at a lower price. According to literature, “Grid computing is a system where multiple applications can integrate and use their resource efficiently”. According to Foster and Kesselman, “A grid is a system that has three important categories: coordination of resources not under centralized control, use standard general purpose interface, and it delivers nontrivial quality of service”. Kon et al define grid computing as, “coordination of resource sharing and dynamic problem solving in multi-institution virtual organizations” [3].

II. SECURITY REQUIREMENTS

Grid systems and applications require standard security functions which are authentication, access control, integrity, privacy, and no repudiation. Authentication and access control issues are. It (1) provide authentication to verify the users, process which have user's computation and resources used by the processes to authenticate (2) allow local access control mechanisms to be used without change. To develop security architecture we have to satisfy the following constraints which are taken from the characteristics of grid environment and application [1].

Single sign-on: A user should authenticate once and they should be able to acquire resources, use them, and release them and to communicate internally without any further authentication.

Protection of credentials: User passwords, private keys, etc. should be protected.

Interoperability with local security solutions: Access to local resources should have local security policy at a local level. Despite of modifying every local resource there is an inter domain security server for providing security to local resource.

Exportability: The code should be exportable i.e. they cannot use a large amount of encryption at a time. There should be a minimum communication at a time.

Support for secure group communication: In a communication there are number of processes which coordinate their activities. This coordination must be secure and for this there is no such security policy.

Support for multiple implementations: There should be a security policy which should provide security to multiple sources based on public and private key cryptography.[8]

III. GRID SECURITY CHALLENGES

Multiple resources provide the control policies to the third party. The VO is one which coordinates the resource sharing and use. The dynamic policies and entry of new participants in the system gives the need for three key functions which are:

Multiple security mechanisms:

Organizations which participate in a VO have investment in security mechanism and infrastructure. Grid security interoperates with these mechanisms.

Dynamic creation of services:

Users must be able to create new services (e.g., "resources") dynamically without administrator permission. These services should coordinate and interact with other services. So, we must be able to name the service with acceptable identity and should be able to grant rights to that identity without any contradiction with the governing local policy.

Dynamic establishment of trust domains:

VO needs to establish coordination between its user and all the resources so that they can communicate easily. These domains must establish trust dynamically whenever a new user join or leave a VO. A user-driven security model is needed to create new entries of the user so that they can coordinate with the resources within the VO.[4]

IV. GLOBUS TOOLKIT SECURITYMODELS

The Globus Toolkit's Authentication and Authorization components provide the basis standard for the "core" security software in Grid systems and applications. Globus software development kits provide programming libraries, Java classes, and essential tools for a PKI, certificate-based authentication system with single sign-on and delegation features, in either Web Services or non-

Web Services frameworks. Grid security technology such as GSI and CAS are used to provide security. These technologies are used to represent the security and are used in various grid projects. Web security services work under the OGSA architecture. It is used to represent refactoring, refinement and repacking of various Grid protocols so that better use of useful resources can be done[3]. OSGA is used with the Globus toolkit to provide WSDL for interface to provide Grid services. OSGA is also used to provide an interface for discovery of grid services. Recent goal of OSGA security work is to provide relationships between OSGA security mechanism and emerging WS security mechanism [9].

A. GT2 Grid Security Model

The security technologies incorporated in the Globus Toolkit version 2 (GT2) includes services for Grid Resource Allocation and Management (GRAM), Monitoring and Discovery (MDS), and data movement (GridFTP). These services use Grid Security Infrastructure (GSI) to provide security. GSI works on a common format based on X.509 identity certificates and a common protocol based on transport layer security (TLS, SSL). An X.509 certificate is associated with private key that forms a unique authentication set that a Grid uses to authenticate itself to other Grid entities.[10]

The TLS-based protocol is used to provide message protection (encryption, integrity checking), according to the requirement of data stream. Gateways are used to translate information between common GSI infrastructure and local site mechanisms. For example, the Kerberos Certificate Authority (KCA) provides an interface for translation of Kerberos to GSI and vice versa.[8]

Each GSI certificate is issued by certificate authority (CA), which runs a large number of organization or commercial company. To trust the X.509 communication, the CA issues the certificate to trust the entity. An X.509 identity certificate is used within GSI for establishment of a trusted communication.[11]

In mechanisms such as Kerberos, where for inter-institutional a bilateral agreement is required atthe organizational level, trust in a CA is established unilaterally: A single entity can decide totrust any CA, without involving the whole organization. This feature is used in the establishmentof VOs in which some portions of the organizations are only used and not the wholeorganization.[12]

GSI introduces X.509 proxy certificates, which is an extension to GSI used by X.509 identitycertificates to allow a user to assign a new X.509 identity to an entity and then delegate subset oftheir rights to that identity. Users create this proxy certificate by issuing a new X.509 certificatesigned by it without involving the CA. By this mechanism new authentication and identities canbe created quickly as there is no involvement of the administrator. To create a trustedcommunication VOs is provided for both the proxy certificate and for security services,Example,the Community Authorization Service (CAS). According to GSI policy if any two entities haveproxy certificates issued by the same user they can trust each other. This policy allows the usersto create trusted communication itself by issuing proxy certificates to any services with whomthey wish to collaborate. [13]

This policy of trust between proxy holders allows then for an easy and simple trust domains but forcomplicated trust domains they have some limitations, for example, limited trust betweenmultiple parties in which we can use security services such as CAS that allow flexible, expressivepolicy to be created for multiple users in a VO. CAS allows a VO to use the policy that has beenprovided to it by the resource providers in the VO. This process has three steps shown in Figure1. The three steps of the figure are:

Firstly, the user authenticates to CAS and receives notification from CAS stating VO's policy thathow the user may use VO resources.

Secondly, after that the user presents the details to a VO resource and the usage request.

Finally, then evaluation is done whether to allow the request, for this the resource checks bothlocal policy and the VO policy expressed in the CAS assertion. CAS allows a resource to retainthe authority over that resource, but it also allows the VO to control the enforced policy. Then,the VO coordinate the policy that how the resources will be shared.

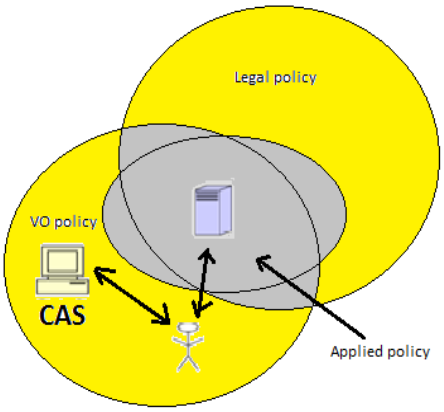

Figure1: Policy of CAS and VO.

While designing GSI we have several efforts to build on PKI. Some of these efforts with respectto Grid security requirements are:

First, Kerberos needs the site administrators for establishment of inter domain trust between newentities.

Second, the CRISIS i.e. a wide area security system provides a uniform and scalable securityinfrastructure in a wide area network but does not provide interoperability with local securitymechanisms.

Third, Secure Shell (SSH) gives us a system of authentication and message protection but doesnot support translation between different mechanisms or creation of dynamic entities.[9]

B. GT3 Security Model

Grid security challenges are solved with Open Grid Services Architecture (OGSA) along with aset of technical specifications to integrate Grid technologies with Web services technologies. Webservices technologies allow defining software component in terms of access methods, to bindthese methods with specific communication mechanisms, and also to provide mechanisms fordiscovering relevant services.

There are no particular mechanisms but few are emerging as ubiquitous. The Simple Object Access Protocol (SOAP) provides an interface for messaging using XML along with HTTP. The Web Services Description Language (WSDL) provides a method for expressing operationsignatures and also bindings to protocols and endpoints in an XML document.

OGSA is a standard Web service interfaces and behaviors to add Web services with the conceptsof crateful services and secure invocation, and also capabilities to address Grid-specificrequirements. These interfaces and behaviors define a "Grid service" and allow users to managethe Grid service's life-cycle, according to the policies, and create sophisticated distributedservices.[6]

A grid service is defined as an interface for service data elements (SDEs) that other entities canquery or subscribe to. OGSA introduces new opportunities and challenges for Grid security.

Globus Toolkit (GT3) and Grid Security Infrastructure (GSI3) were the first to implement OGSAmechanisms. GT3's security model allows applications and users to operate on the Grid as easyand automatic manner as possible. Security

mechanisms should not be instantiated in anapplication but should be supplied by the surrounding Grid infrastructure to adapt on behalf of theapplication to meet the application's requirements.[7]

The application should deal only with application specific policy. GT3 uses the following features of OGSA and Web services security to achieve their goals, these goals are to:

First, use of security functionality as OGSA services to locate them and use the service whenever needed.

Second, use of sophisticated host environment to provide security for applications and to adapt security of application without changing it.

Third, to publish service security policy for clients to discover dynamically what are the requirements and mechanisms needed for establishing trust with the service.

Fourth, to provide specifies standards for the exchange of security tokens for interoperability.[4]

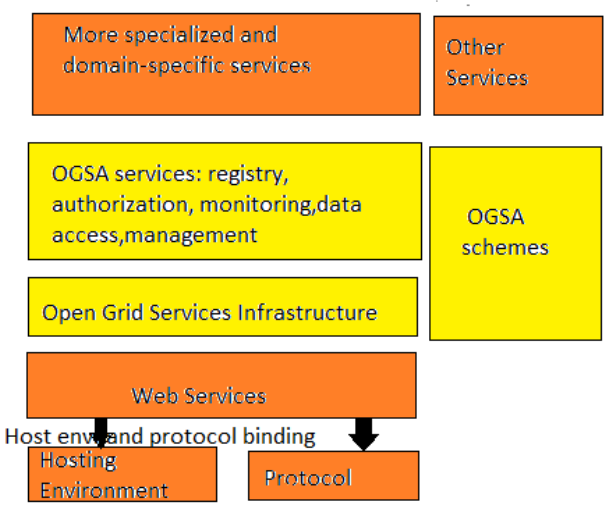

Figure2: OGSA Architecture

C. GT4 Security Models

GT4 Authorization implements SAML (security assertion markup language), and uses theXACML (extensible access control markup language). XACML authorization frameworkarchitecture is an implementation of the Open Grid Services Architecture is an initiative forrecasting Grid concepts within a service oriented framework based on Web services.[8]

In GT4, we have additional Web Services security specifications implementation. Web Serviceshas provided several security standards that which influences Grid computing. XACML andSAML are the two important authorization standards. We have several other authorizationsystems that support Grid computing that are Akenti, PERMIS, Shibboleth and VOMS. Akenti,

PERMIS and Shibboleth use the type of attributes which are needed to make authorizationdecision. VOMS provides user attributes used for authorization. These authorization systems havetheir own policies, and can be integrated with GT4 authorization framework to provideauthorization services.[5]

The XACML authorization model has the following policies that are used to createcommunication. The functioning of these policies are:

Firstly, The PEP (Policy Enforcement Point) is used to accept the access requests from users andthen it sends the requests to the PDP.

Then, The PDP(Policy Decision Point) then makes the access decisions according to the securitypolicy or policy set of PAP (Policy Administration Point) and also by using attributes of thesubjects, the resource, and the environment that are obtained by querying the PIP (PolicyInformation Point).

After that, the access decision taken by the PDP is then sent to the PEP. Finally, according to thedecision of PDP, the PEP then either permits or denies the access request.[9]

The whole architecture is shown below.

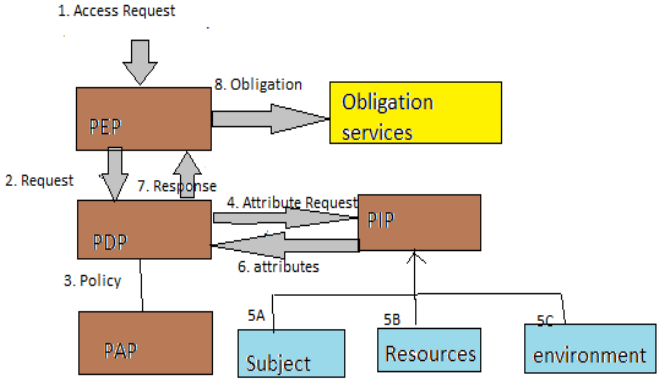

Figure3: XACML authorization model

XACML also defines a policy language. Policies are organized in hierarchy with the Policy Setscombined using combining algorithms. A rule is has a target, an effect of that rule and a conditionon which the rule works. A Policy comprises of a target, one or more rules, and an optional set ofobligations.[9]

V. SMARTGRID CYBER SECURITY

The cyber security for the smart grid is the possibility that if in a centralized grid we have twoway digital communications the grid can become susceptible to the hackers who can use customerconfidential information and can cause adverse effect on the communication. This is the latestconcern in the Grid to create a Smart Grid cyber security to provide the internet security in theGrid. There should be some policies with which we can take the benefits of the Internet and alsothe available computation power in a secure way.[7]

Internet facility is much more reliable than electric grid due to the following reasons:

1) Internet is decentralized and is in starfish pattern and not in spider,
2) Asynchronous i.e. we don't have to use a single source we can work on different server and
3) It has many paths and not a few single connections. The Internet is a smart grid, a resilientgrid, a self-healing grid that does not go down. The last connection to the grid may fail, or aparticular destination may fail.

The Internet makes it possible to have a more secure grid as it reliably monitor and control everypart of the grid in real time. The new smart grid will be a less centralized grid because:

1) the traditional economies that supported it has been removed by risk and uncertainty of sitting,construction, operation, fuel supply, environmental impact and cost recovery,
2) There is penetration of distributed generation, storage, PHEVs/EVs as well as customerpremises energy management systems
3) There is an increasing penetration of stochastic, energy sources like wind, solar and consumerdispatched generation. There is a complex grid with many points to automatically monitor andcontrol resources.

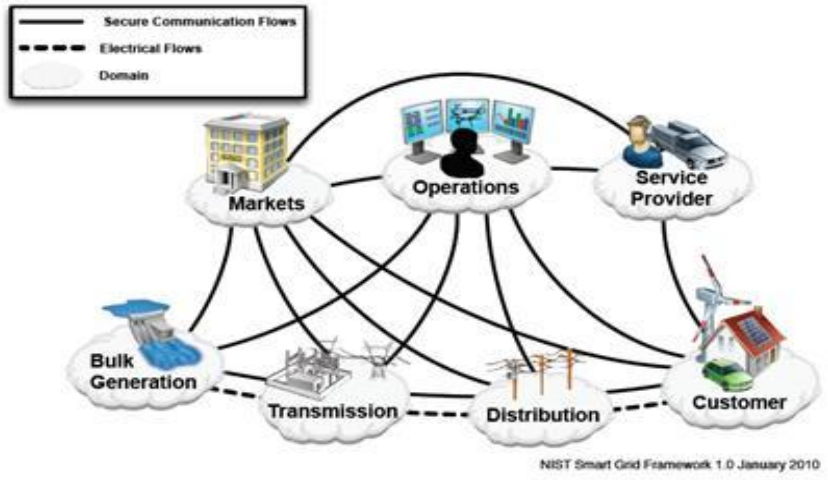

Figure 4- Cyber Grid

A.Why Cyber Security Is So Hard For The Future Grid

The following reasons due to which the cyber security will not be implemented so easily in futuregrid are:
First is, Legacy controllers, networks Fragile security, built to run on private data links, 24/7(hard to update, patch), real time requirements (security, crypto may impact timing).
Second, Control nets run over (or tunneled though) public networks (attack channel, or subject tobroader disruption).
Third, Best Practices for Control Systems & Grid Security (DHS, NERC CIP standards, NIST
Draft NISTIR 7628, etc.).
Fourth, these are processes for developing secure systems, not cookbook answers!
Fifth, Security is a system issue—what are the pieces and how do they work together
And the last is- Security is a moving target.[2]

VI. DESCRIPTION OF NEWLY CONSTRUCTED AUTHENTICATION SCHEME

To apply the proposed authentication scheme we have classified Grids into two distinct nodes, namely, A. Administrator Grid Node (AGN) & B. Compute Grid Node (CGN). A user in CGN requests AGN to process the authenticator operation. After checking, AGN sends acknowledgement to CGN specifying whether the user can continue the process or not.

A. Administrator Grid Node (AGN)

To process the user's request AGN needs some tables & operations. To store user information & compare it with the Grid information, AGN needs one database, which is named as Data Base for Generating User Authenticator (DBGUA). The required tables are as follows:

Table: Receive Request: It is used to match the authenticator for the existing users.

Request Command	User-Id	Authenticator

Table: DBGUA: It is used to store the authenticator (produced by the encryption algorithm) for each user.

User-Id	Authenticator

Table: New User Request: This table is used to produce new authenticator for new users. The operation is called new user operation.

New	User-Id	Password	Authenticator

Table: Update User Request: If any user sends update request to AGN for changing his authenticator, AGN uses different attributes to check for authenticator & then if the verification produces no error, it updates the existing authenticator of the user with a new one.

Update	User-Id	Old Password	Old Authenticator	New Password	New Authenticator

Table: Delete User Request: If any user sends delete request to AGN for removing his authenticator, AGN uses different attributes to check for authenticator & then if the verification produces no error, it deletes the existing authenticator of the user.

Delete	User-Id	Password	Authenticator

Table: User Permission Request: If any user needs to access his authenticator, it sends permission request to AGN for getting the chance for accessing.

Permission	User-Id	Password	Authenticator

Table: Decryption Request: If any user sends decryption request to AGN for decrypting contents, AGN uses different attributes to check for authentication & then if the verification produces no error, it decrypts the content.

User-Id	Password	Content	Decryption

B. Compute Grid Node (CGN)

The user resides in the CGN sends requests to AGN for processing some operations. The acknowledgements received from the AGN are stored in the local database of CGN, namely Compute User Information Data Base (CUIDB).

Table: CUIDB:

User-Id	Authenticator

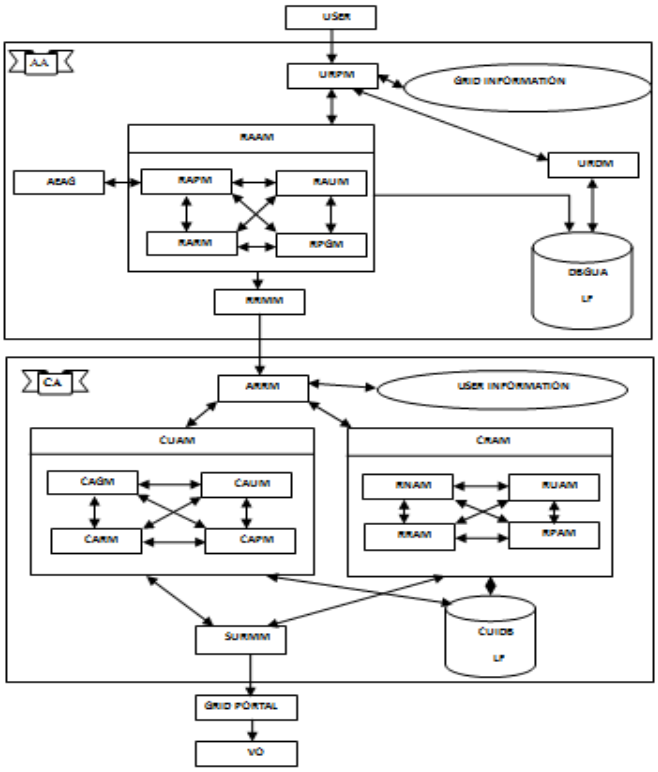

Fig.: Architecture of Newly Constructed Authentication Scheme

VII. THE ARCHITECTURE FRAMEWORK OF NEWLY PROPOSED AUTHENTICATION SCHEME

The acronyms used in the above figure, are described below:

AA: Administrative Authentication: The authentication process used in Administrator Grid Node.

CA: Compute Authentication: The authentication process used in Compute Grid Node.

URPM: User Request Process Module: It resides within AGN & processes the user requests from CGN using the following modules.

RAAM: Remote Authenticator Access Module: It helps user to request for accessing authenticator using the following sub modules.

RAPM: Remote Authenticator Producer Module: It produces authenticator for users. It calls **Authentication Encryption Algorithm Generator (AEAG)** to produce authenticator for a particular user.

RAUM: Remote Authenticator Updater Module: It updates the existing authenticators as per needs.

RARM: Remote Authenticator Remover Module: It deletes the corresponding authenticator if an authentic user wants to delete that.

RPGM: Remote Permission Granting Module: This module is used to provide permissions to access resources as per needs.

RRMM: Returning Result Message Module: It returns the result message to the CGN.

URDM: User Request Decryption Module: This module is used to decrypt data on user request.

ARRM: Administrator Request Response Module: It returns the user's information as per the needs of AGN.

CUAM: Compute User Authentication Module: It has following four modules:

CAGM: Compute Authenticator Generate Module: It sends new user request to AGN to produce authenticator for the same.

CAUM: Compute Authenticator Update Module: It sends update request to AGN to update existing authenticator.

CARM: Compute Authenticator Remove Module: It sends delete request to AGN to delete existing authenticator.

CAPM: Compute Authenticator Permit Module: It sends permission request to AGN to get permission to access procedures & databases.

CRAM: Compute Receive Authentication Module: It also has the following four modules:

RNAM: Receive New Authenticator Module: It receives new authenticator produced by the new user request & sends it to be stored in the CUIDB.

RUAM: Receive Updated Authenticator Module: It receives the updated version of the existing authenticator.

RRAM: Receive Remove Authenticator Module: It receives the authenticator to be removed.

RPAM: Receive Permit Authenticator Module: It receives the authenticator who got the permission.

SURMM: Send User's Request Message Module: Finally the produced authenticator is sent to the Grid portal.

VIII. AUTHENTICATION ENCRYPTION ALGORITHM GENERATOR

Related Work: Nirmalya et al[13] presented some algorithms to enhance the work done by Lee et al, where they solved some security issues. We, here will modify the existing solutions & will provide better solutions for user authentication.

Concept: When a user enters his\her User-Id & Password, it comes as plain text; but for security reasons we will convert it into Cipher text. To do so, we must perform the following things:

1. Change the contents of plain text to cipher text;2. Locking Protocol;3. Position Exchange;4. Network Transmission;5.Data Uncertainty; 6. Produce Authenticator.

Modified Algorithms:

1. Change the contents of plain text to cipher text:

In the existing algorithm of Nirmalya et al[13], they described the algorithm as follows:

Input: plaintext (PT) (User-Id=UI,Password=PW), input by the user.

Do: for N users,

Create a symbol table (ST) to store Plain Text (PT) as $U_1P_1U_2P_2U_3P_3U_4P_4.....U_NP_N$

Where N=U+P,

For all $1 \le i \le N$,

Set a key of string by taking

$KS = U_{i+1}P_{i+2}$ up to N times

In the above code, number of KS generated would be twice of the number of users, since N=U+P. The key String generated are also redundant.

Thus to normalize the problem of redundancy, N can be taken as N = (U+P)/2.

The modified algorithm is as follows:

Input: Plaintext (PT), User ID = UI, Password= PW

Do: for N users,

Create a Symbol Table (ST) to store PT as $U_1P_1U_2P_2U_3P_3$…….U_NP_N, where N= (U+P)/2

For all $1 \le i \le N$,

Set a key of string by taking

$KS = U_{i+1}P_{i+2}$ up to N times

Extract the ASCII values of username and password from RAM

Perform XOR operation as

$U_i'' = C'(U_i')$, where C' is the 9's compliment operation, & U_i' is defined as $U_i' = U_i + U_{i+1}$

Get the cipher value CU_i'' for the same as:

$CU_i'' = U_i \%26$

Consider this CU_i'' value as ASCII of a particular semantics. Replace this with previous one.

Repeat this for password to get CP_i''.

END Do.

2. Locking Protocol:

Nirmalya et al[13] generated Key Strings in Authentication Encryption algorithm but was left unused. Here we have used those Key Strings to lock the Cipher text. This increases the complexity of the string and hence security level increases.

The algorithm is as follows:

Input:

CT= Cipher text, KS= Key String

Find the length of the Cipher text (CT).

Get the strings CT and KS into arrays.

Repeat for i=1 to twice of the length of CT

Check for odd position

If odd position found, insert CT element.

Else, insert KS element.

3. Position Exchange Algorithm:

Nirmalya et al[13] used a position shifting algorithm, where the elements were shifted by one position linearly. To increase the security we have proposed a new position exchange algorithm. Here the elements are exchanged between the columns diagonally such that the first element first column would be exchanged with the last element of next column.

The algorithm is as follows:

Initialize:

n=no. of rows

m= no. of columns

k=0 and a temporary variable, temp

Repeat for i=n to 0

Repeat for j=0 to m

Swap the position of array[k][j] with array[i][j] using temp.

Increment k by 1.

Repeat for i=0 to n

Repeat for j=0 to m

Swap the position of array[i][j] with array[i][j+1] using temp.

4. Network Transmission:

Nirmalya et al used TFRC protocol for Network Transmission. But TFRC is used only in unicast environment. Since, grid computing is a multicast environment, so a protocol supporting multicast environment must be used for transmission through the network. Therefore, TFMCC (TCP Friendly Multicast Congestion control) can be used which extends the basic mechanisms of TFRC's to cope with multicast connection.

Mathematical Modeling of TFMCC Protocol:

The throughput (X) of clients is defined by,

$$X = \frac{8 * S}{R * \{\sqrt{\frac{2P}{3}} + 12 * \sqrt{\frac{3P}{8}} * P * (1 + 32P^2)\}}$$

Where, X is the transmit rate in bits/second,

S is the packet size in bytes,

R is the Round-Trip Time (RTT) in seconds and

P is the loss events as a fraction of the no. of packets transmitted.

R can be calculated as:

R= TR_now – TR_*R'*

Where, TR_now is the time the data packet arrives at the receiver, and

TR_R' is the receiver report time-stamp echoed in the data packet.

The data loss can be calculated by the equation at a low-pass filter as:

$$Y_i = WY_{i-1} + (1-W)X_i$$

Where, W is consider as a constant & its value is 65000/65536, which corresponds to acorner frequency of approximately 0.0013 $packets^{-1}$

The loss interval can be calculated from equation 1 as:

$$\text{X\_recv} = \sqrt{\frac{3}{2}} * \frac{8*S}{R*\sqrt{\frac{1}{L\_o}}}$$

$$\text{Therefore, L\_o} = \{\frac{X\_recv*R}{(\sqrt{3/2})*8*S}\}^2$$

Receivers can adjust L_o by the following equation:

$$L\_o = L\_o*(R/R\_\max)^2$$

The expected delay until the first feedback message is sent is:

$$E[D] = \frac{T}{\log N}\int \frac{(1-x)^n}{x}dx \approx T(1-\log_N^n)$$

The expected number of feedback message can be calculated as:

$$E[M] = N^{\lambda/T}\{\frac{n}{N} + (1-\frac{1}{N}) - (1-\frac{1}{N^{\frac{\lambda}{T}}})^n\}$$

Where n is the actual no. of receivers

The following exponentially weighted random timer mechanism sets up the feedback timer to expire after

$$t = \max(T*(1+\log(x)/\log(N)),0)$$

Where,

X is a random variable uniformly distributed in (0, 1),

T is the duration of feedback round (i.e., 6*R_max), and

N is the estimated upper bound on the no. of receivers.

5. Data Uncertainty:

Data uncertainty refers to the discrepancy between data and the spatial characteristic represented by the data. It is the quantitative estimation of error present in the data. It is generated by either systematic error, or random error or statistical error. This uncertainty affects the reliability of applications using the data. Careful methodology can reduce uncertainty by correcting for systematic error and minimizing random error. However, uncertainty can never be reduced to zero. In Authentication, correlated uncertainty is generated because the User-Id & Password is dependent to each other. We will solve data uncertainty problem using cumulative joint probability distribution.

Case 1. If both the User-Id & Password are discrete random strings, we will use Probability Mass Function (PMF). So,

$$PMF(U=u \,\&\, P=p) = \begin{cases} PMF(P=p \mid U=u).PMF(U=u) \\ PMF(U=u \mid P=p).PMF(P=p) \end{cases}$$

According to the rules of probability, we can write,

$$\sum_u \sum_p PMF(U=u \,\&\, P=p) = 1$$

Case 2. If both the User-Id & Password are discrete random strings, we will use Probability Density Function (PDF). So,

$$PDF_{U\,P}(u,p) = \begin{cases} f_{P|U}(p \mid u)\, f_U(u) \\ f_{U|P}(u \mid p)\, f_P(p) \end{cases}$$

Again, according to the rules, we can write,

$$\int_U \int_P PDF_{UP}(u,p)\,du\,dp = 1$$

6. Produce Authenticator:

Authentication is the Process of giving individuals access to system objects based on their identity.It merely ensures that the individual is who he or she claims to be. Authenticator is an object that knows how to obtain authentication for a network connection. The algorithm is as follows:

Input: User-Id & Password
Do: Get the hostname & host address of the Authentication Server or NULL if not available.
Get the port number for the requested connection.
Get the prompt string given by therequestor.
Get the Authentication scheme for the connection.
Set the protocol for the requestedconnection.
End Do.
Output: Generate the Authenticator for the current user who had entered User-Id & Password.

IX. EXPERIMENTAL RESULTS

After constructing the algorithms, we have created a grid environment through GridSim, BOINC and Globus Toolkit (Version:6.0) and applied those. The results are as follows:

Table 1: The plain text equivalent cipher text:

Execution	Alphanumeric Codes	Cipher Text
1	A	5A*VWI
2	B	IR)6PK
3	C	FC!3&H
4	D	KE0^@N
5	E	Y2&K9T
6	F	$UR#Q0
7	G	A2(K4~H
8	H	K2`S@S
9	I	&F2M!L
10	J	G)3#Q*
11	K	U1B|MV
12	L	X+S7P!
13	M	V6)2@G
14	N	J|EL%3
15	O	U1)K6D
16	P	>WX!7G
17	Q	%G2KO5
18	R	ZM2$Q>
19	S	7F1H~K
20	T	L1%5K*
21	U	5)GH`S
22	V	QX8O0@
23	W	M23:I<
24	X	“WX00!
25	Y	71YZ#K
26	Z	>KL0’H
27	0	#S>4K”
28	1	TR^7)K
29	2	@GH3~K
30	3	4=DM%8
31	4	2)A#K0
32	5	M1T&IO
33	6	9L%X)3
34	7	OP7*V`
35	8	@FR26”
36	9	$KQ5’J

Table 2: Cipher text after using Locking Protocol:

USER ID	PASSWORD	KEY STRING	CIPHER TEXT OF USER ID	LOCKED TEXT OF USER ID
A1	P1	B2R3	A#2)DR3UI@O'	AB#22R)3DR3UI@O'
B2	Q2	C3P1	KN5%JO62$FB	KCN35P%1JO62$FB
C3	R3	A1Q2	Q2#ZS*2K)7VM	QA21#QZ2S*2K)7VM

Table 3: Time complexity comparison between the existing algorithm (proposed by Nirmalya et al) and our newly developed algorithm:

As per the above table, if we consider a string like "TIGER1105", the equivalent cipher text will be "F G6A A Y! ^ V3(1S^ N E2A9I %7=&S$*@:$4S@,0J TX2(A5^4S@ TX^" which is very difficult to break.

The following graphs show the comparison between the algorithms more easily.

Execution	Length of user-id & password in bits	Length of key string in bits	Time required for existing algorithm in ms	Time required for proposed algorithm in ms
1	12,15	22	0.00378	0.00371
2	10,08	29	0.00417	0.00413
3	13,12	22	0.00377	0.00375
4	07,16	20	0.00369	0.00360

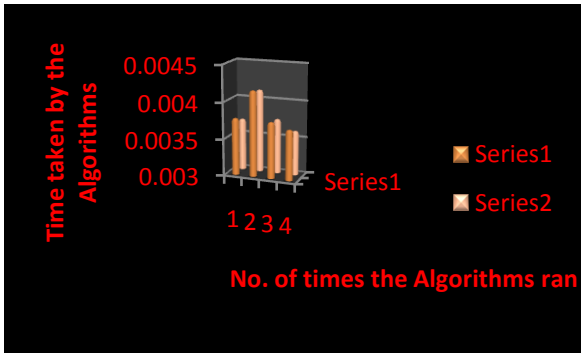

Figure 1:

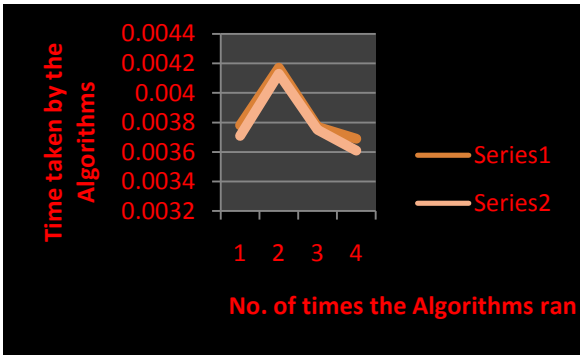

Figure2:

In the above graphical representation we can observe that the time complexity of our newly proposed algorithm is lower than the time complexity of the existing algorithm. Hence it is providing better throughput to grid environment.

X. CONCLUSION

This paper is basically an extension of our previous paper. In this paper we have introduced a robust authentication mechanism at the entry points of Grid. After executing and implementing the respective codes for our proposed algorithms, we find that our algorithm is providing a bit better time complexity rather than the existing one. In this paper, we have provided an advanced architecture with algorithm for secure authentication process and experimental results showing better output in context with robustness for data security in Grid.

REFERENCES

[1] I. Foster and C. Kesselman, "The Grid: Blue print for a new computing infrastructure", Morgan Kaufmann Publications (1999).

[2] I. Foster, C. Kesselman, J. M. Nick and S. Tuecke, "The physiology of the Grid: An open grid services architecture for distributed systems integration", Grid Forum white paper, 2003.

[3] Volker Sander, "Networking Issues for Grid Infrastructure", GFD-I.037, Nov, 22, 2004.

[4] I. Foster, C. Kesselman, C. Lee, R. Lindell, K. Nahrstedt, A.Roy. "A Distributed Resource Management Architecture that Supports Advance Reservations and Co-Allocation", Intl Workshop on Quality of Service, 1999.

[5] I. Raicu, Y. Zhao, C. Dumitrescu, I. Foster, M. Wilde. "Falkon:a Fast and Light-weight tasKexecutiONframework",IEEE/ACM SuperComputing 2007. International Journal of Scientific and Research Publications, Volume 3, Issue 8, August 2013 4 ISSN 2250-3153, www.ijsrp.org

[6] The Globus Security Team. "Globus Toolkit Version 4 Grid Security Infrastructure: A Standards Perspective," Technical Report, Argonne National Laboratory, MCS, 2005.

[7] I. Foster, C. Kesselman. "Globus: A Metacomputing Infrastructure Toolkit", Intl J. Supercomputer Applications,11(2):115-128, 1997.

[8] B. Allcock, J. Bester, J. Bresnahan, A. L. Chervenak, I. Foster,C. Kesselman, S. Meder, V. Nefedova, D. Quesnal, S. Tuecke."Data Management and Transfer in High Performance Computational Grid Environments", Parallel Computing Journal, Vol. 28 (5), May 2002, pp. 749-771.

[9] J. M. Schopf, I. Raicu, L. Pearlman, N. Miller, C. Kesselman, I.Foster, M. D'Arcy. "Monitoring and Discovery in a Web Services Framework: Functionality and Performance of Globus Toolkit MDS4", Technical Report, Argonne National Laboratory, MCS Preprint #ANL/MCS-P1315-0106, 2006.

[10] N. Karonis, B. Toonen, and I. Foster. MPICH-G2: A Grid- Enabled Implementation of the Message Passing Interface.Journal of Parallel and Distributed Computing, 2003.

[11] I. Foster, C. Kesselman, L. Pearlman, S. Tuecke, and V. Welch."The Community Authorization Service: Status and Future," In Proc. of Computing in High Energy Physics (CHEP), 2003.

[12] Introduction to Grid Computing by Bart Jacob,Michael Brown, Kentaro Fukui, Nihar Trivedi.

[13] N. Mukhopadhyay, A. Bhowmick, A. Bandyopadhyay. "A Proposed Robust Authentication Approach for Secure Data Transmission in Grid Environment", IJCT, 2013.

